HEP-CCE: Analysis of Root I/O in HEP Workflows with Darshan

Shane Snyder (ANL), Doug Benjamin (ANL),
Patrick Gartung (FNAL), Ken Herner (FNAL)

U.S. DEPARTMENT OF ENERGY | Office of Science
Argonne NATIONAL LABORATORY
BROOKHAVEN NATIONAL LABORATORY
Fermilab
BERKELEY LAB Bringing Science Solutions to the World

Motivation

- Modern HEP workflows must manage increasingly large and complex data collections
- HPC facilities are being employed to help meet the growing data processing needs of these workflows and to reduce the time required to make new scientific insights
- An ability to instrument the I/O behavior of HEP workflows can be critical to characterizing and improving their usage of HPC storage resources
  - The recent shift of HEP workflows to HPC facilities points to potential untapped I/O tuning opportunities

Darshan background

- Darshan is a lightweight I/O characterization tool that can capture either condensed views or entire traces (DXT) of application I/O behavior

- Darshan is deployed at a number of DOE computing facilities (including ALCF, ANL LCRC, and NERSC) and has become a popular tool for HPC users to better understand their I/O workloads

- While originally designed specifically for MPI applications, in the past year we have modified Darshan to also work in non-MPI contexts
  - Extends I/O instrumentation into exciting new HPC contexts, like HEP workflows that have traditionally not been based on MPI

Darshan for ROOT I/O activity status

Integration of Darshan into HEP workflows like ATLAS, CMS, and DUNE can provide deeper understanding of their use of HPC storage resources. This understanding can be used to optimize workflow usage of ROOT as well as to make general improvements to the ROOT I/O library.

- Phase 1: Preparation
  - Introduction to Darshan, ROOT, and HEP workflow (ATLAS, CMS, DUNE) technologies and discussion of potential deployment strategies
- Phase 2: Prototyping
  - Deployment of Darshan for use in different HEP workflows (e.g., in containers)
  - Development of fork-safe Darshan instrumentation strategies (needed for ATLAS)
  - Development of Darshan analysis tools for understanding workflow I/O behavior

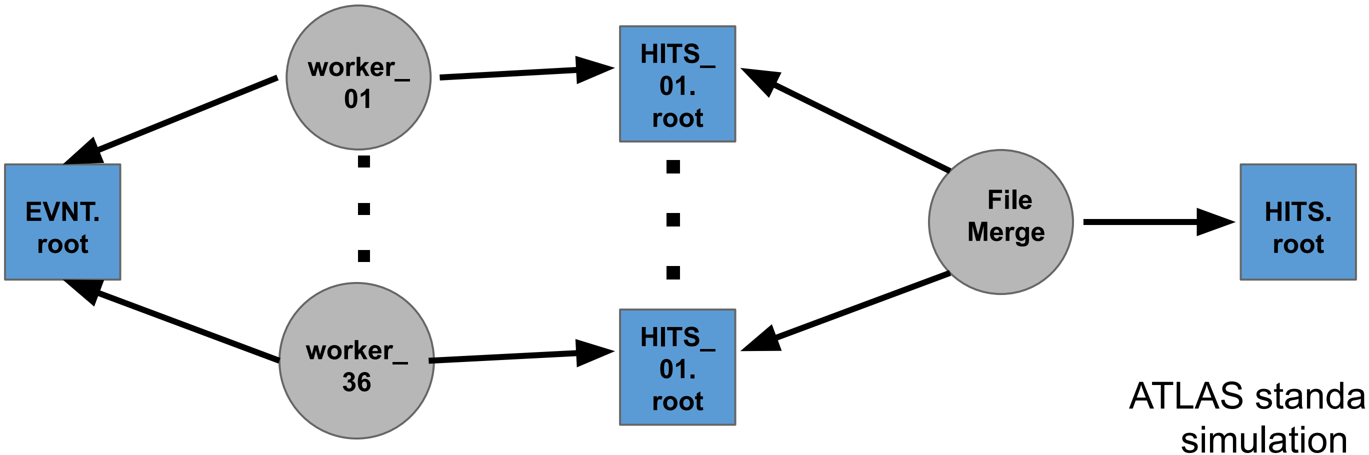
Darshan analysis of ATLAS

Darshan analysis of ATLAS I/O

- ATLAS uses software containers (Singularity and Shifter) to run Geant4 Full simulation on HPC's.
  - Geant 4 Full Simulation is primary workflow on US HPC's
- Used ANL LCRC HPC for this work.
- Rebuilt ATLAS official Singularity image to add hooks for Darshan.
- Built the Darshan Run Time library on shared file system using ATLAS image.
- Had to test interactively because Darshan did not run nicely inside the container with Singularity shell command -
- Tested Standard Simulation (1000 events simulated - 1 output file)
  - With and without DXT traces
- Tested Event Service Simulation (fine grained simulation)
  - With and without DXT traces

Darshan analysis of ATLAS I/O

Atlas AthemaMP - simulates 1 event - forks into many subprocesses (1 per core), each subprocess (worker) process a number of events (up to 1000 events for all workers), After all workers are finished, Serial File merge step to produce 1 output HITS file (ROOT format).

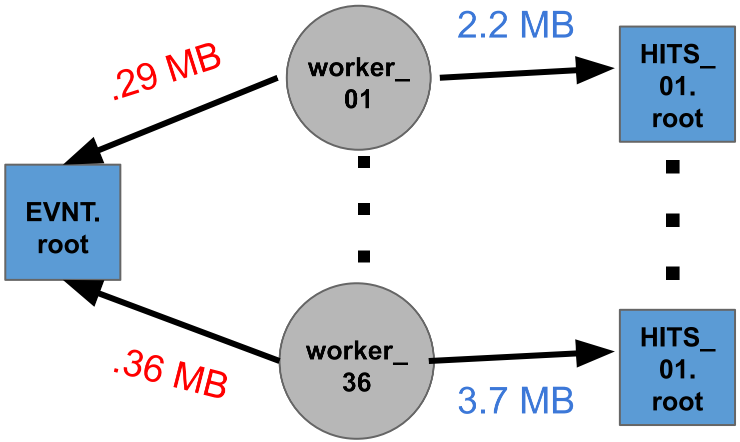


ATLAS standard simulation

Darshan analysis of ATLAS I/O

Each worker reads in the range of .28 - .47 MB of EVNT data

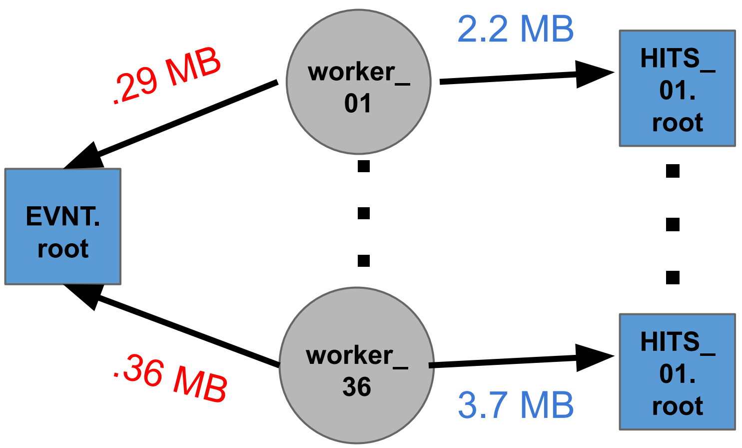
NOTE: worker processes are forked, which Darshan can't reliably instrument, currently

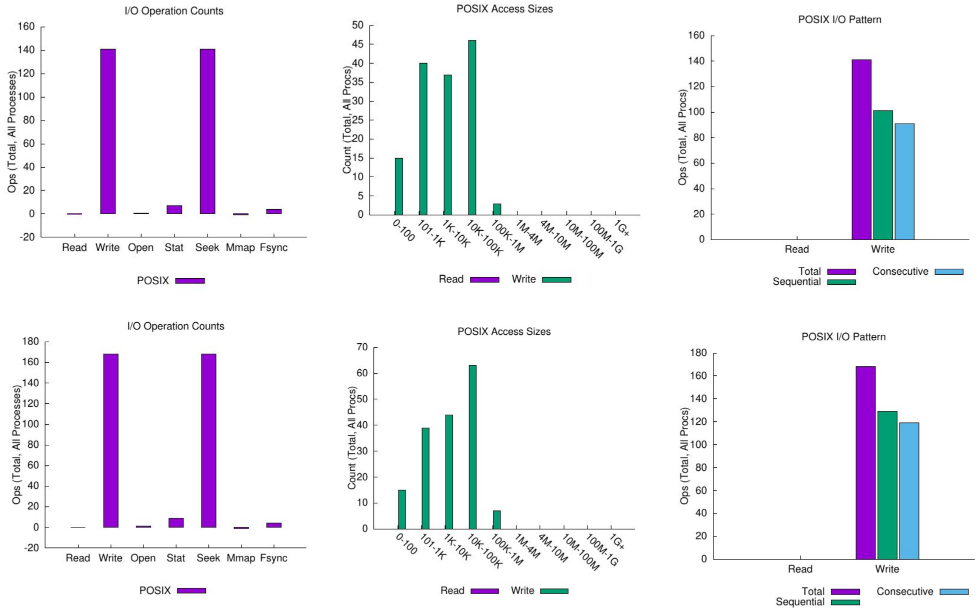


Each worker writes in the range of 2.0 - 5.5 MB of HITS data

Atlas AthemaMP - simulates 1 event - forks into many subprocesses (1 per core), each subprocess (worker) process a number of events (up to 1000 events for all workers). **Geant 4 Full Simulation** run by each worker sub-process.

Darshan analysis of ATLAS I/O



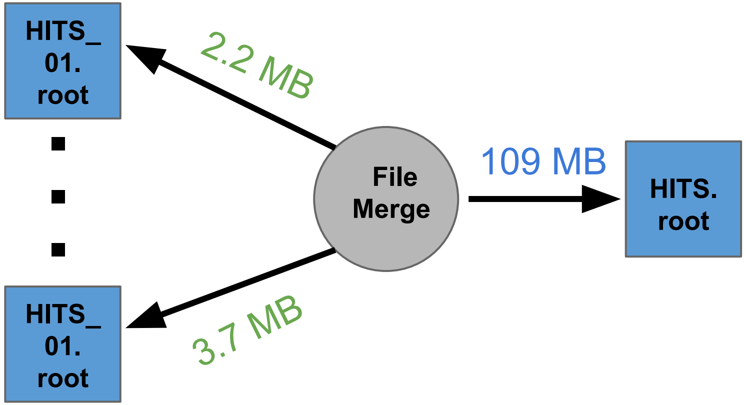


Equal number of writes/seeks, writes mostly sized at O(10s of KB), I/O mostly sequential

Darshan analysis of ATLAS I/O

Workers' HITS files are read serially (i.e., worker1, then worker2, and so on)

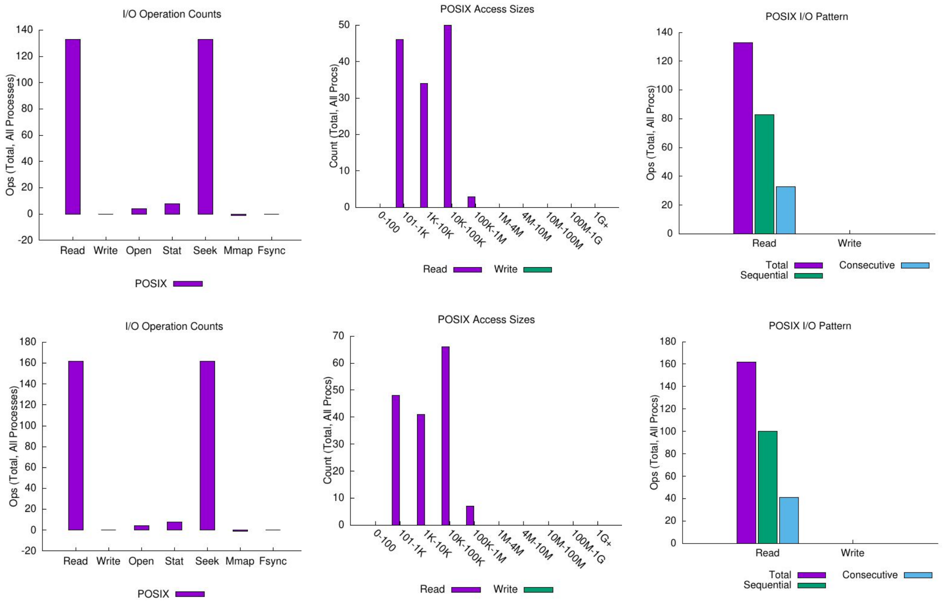
Each worker reads ~30 KB additional data (compared to write phase), yielding 112 MB total read volume

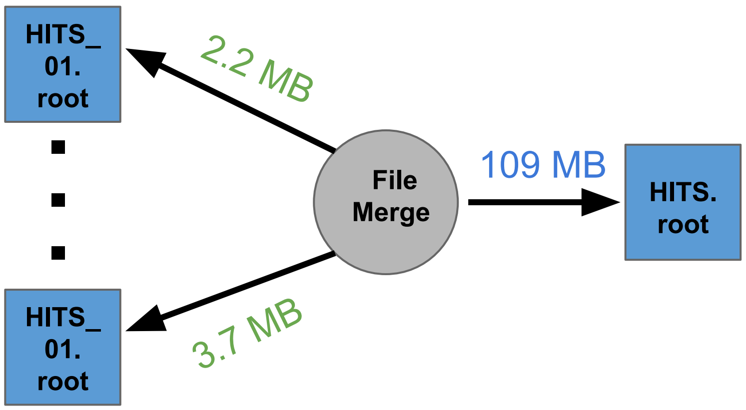


Serial File Merge - 1 output file 1000 events

As input HITS files are read, they are written (merged) into a single output HITS file

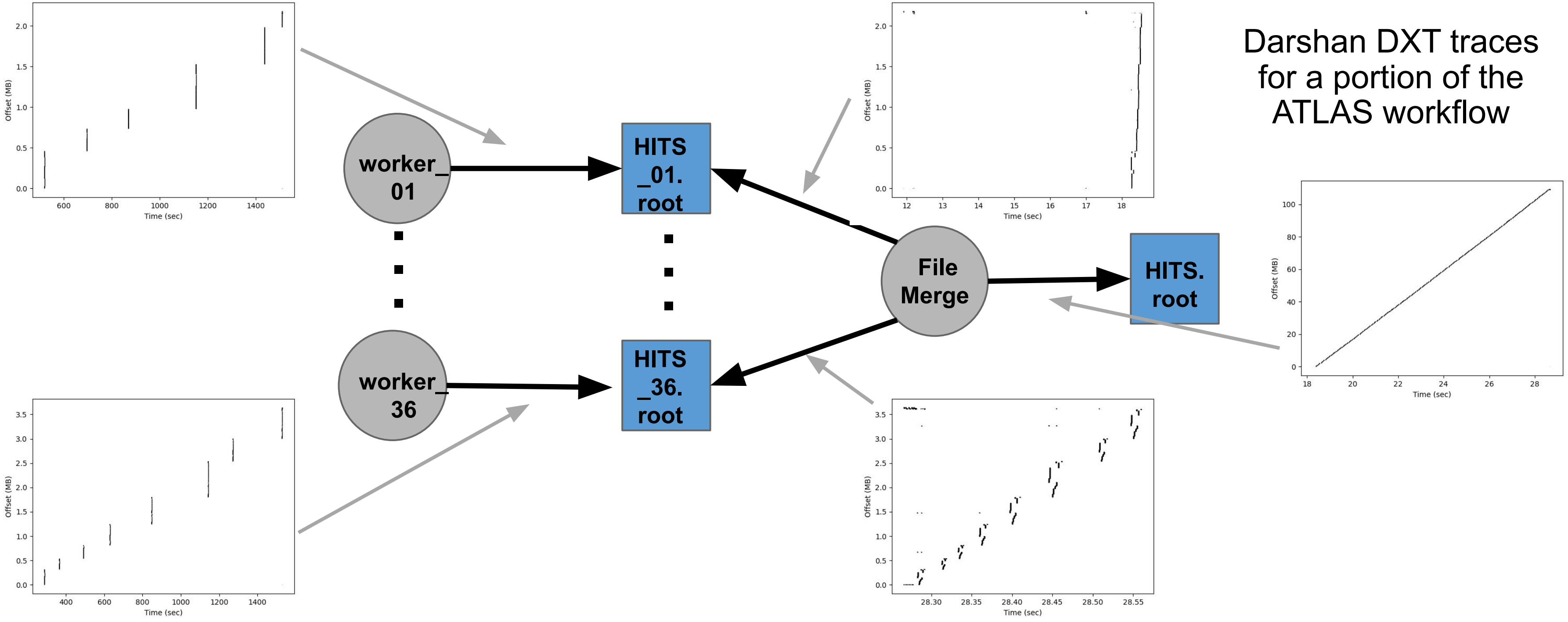
Darshan analysis of ATLAS I/O





Equal numbers of reads/seeks, reads mostly sized at O(10s of KB), I/O noticeably less sequential than write phase

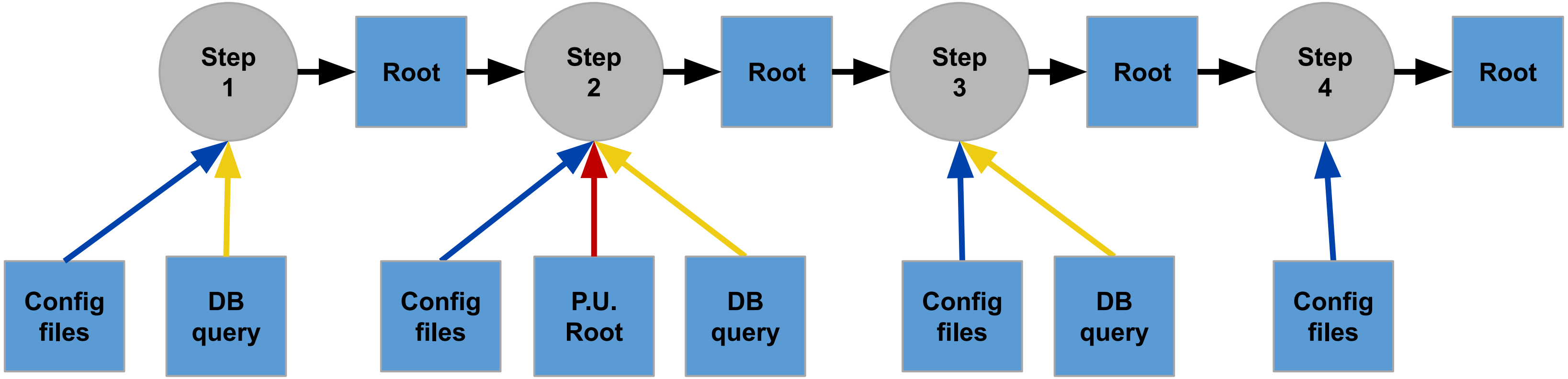
Darshan analysis of ATLAS I/O



Darshan DXT traces for a portion of the ATLAS workflow

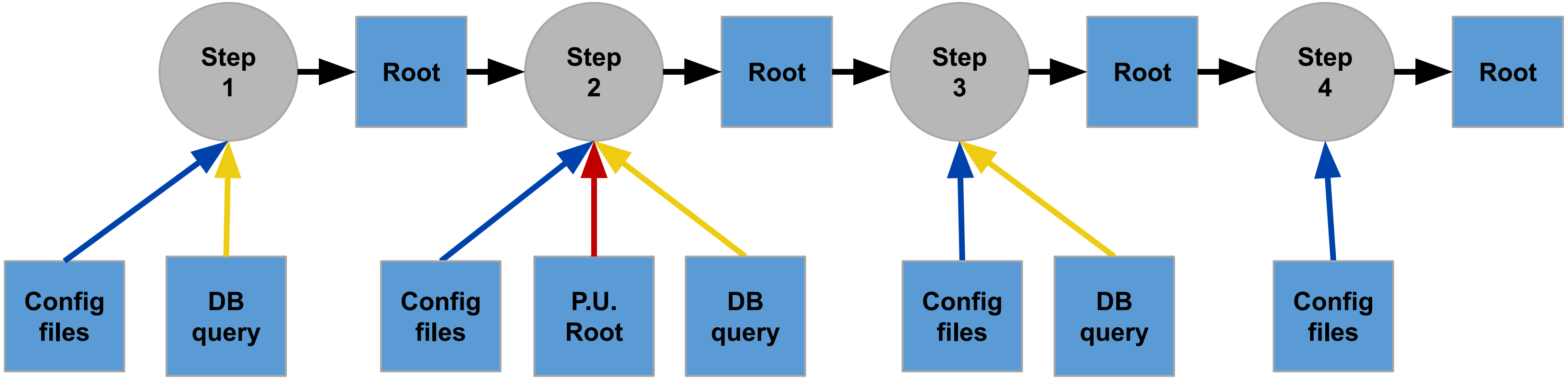
Darshan analysis of CMS

CMS workflow I/O overview



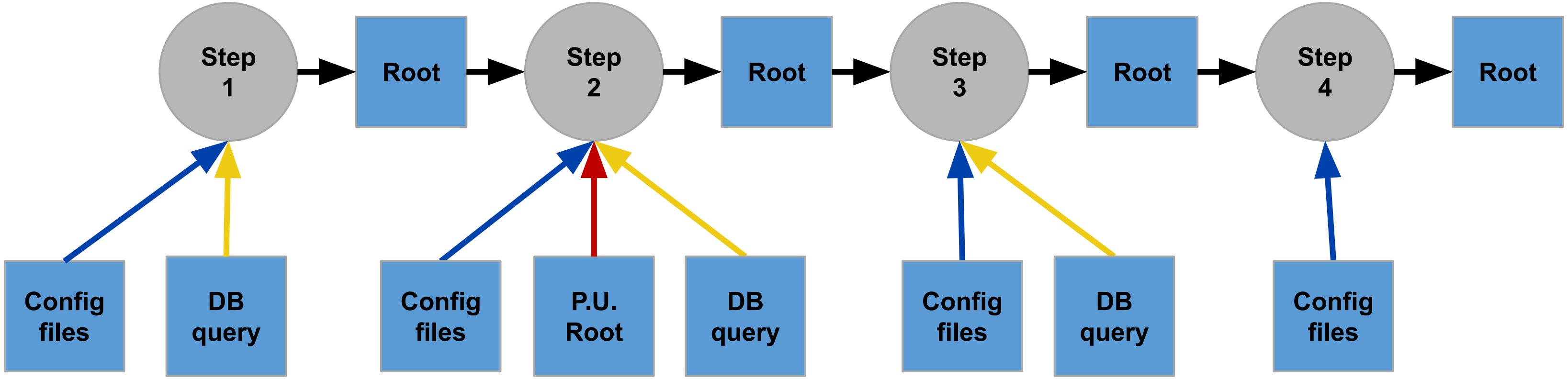
Various data processing stages and data products for a CMS workflow based on event generation, detector simulation, event reconstruction, and subsequent analysis

CMS workflow I/O overview



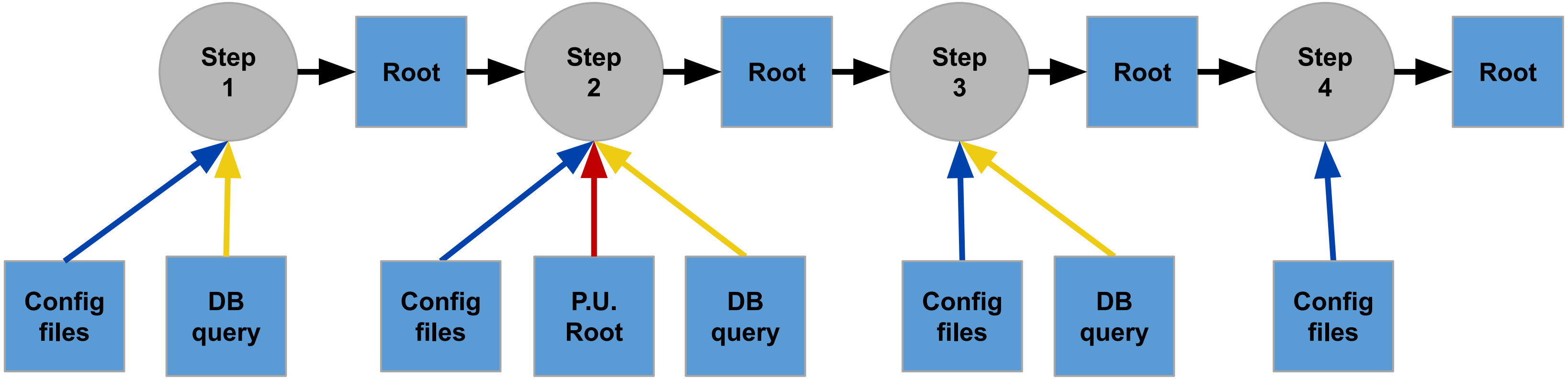
- Config files : Read from local or network disk or cvmfs (blue arrow)
  - cvmfs: Read-only file system that queries files over HTTP and keeps a local cache on disk and is accessed via FUSE mount

CMS workflow I/O overview



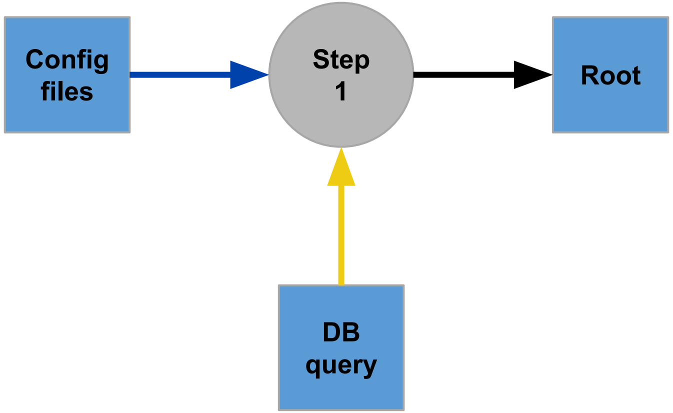
- DB queries: Read detector conditions from database over https (yellow arrow)

CMS workflow I/O overview



- ROOT files:
  - Read/write from/to local or network disk (black arrow)
    - NB: all measurements made with local disk
  - Or read via XRootD or EOS protocol (red arrow)
    - XRootD: Protocol for streaming reads of ROOT files
    - EOS: Protocol with POSIX reads of ROOT files

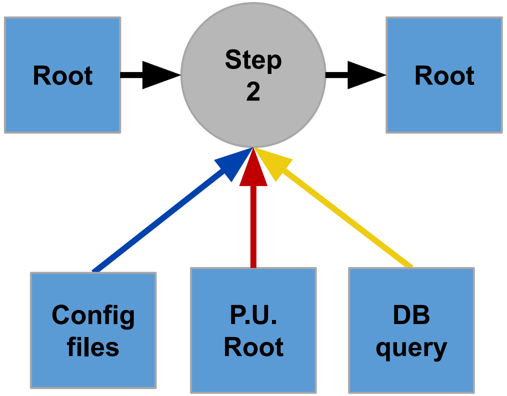
CMS workflow I/O: Step 1 (GEN/SIM): Generation and simulation



Generation of events and simulation of interactions with detectors

- Inputs: config files, DB query
- Outputs: Simulated hits in ROOT file

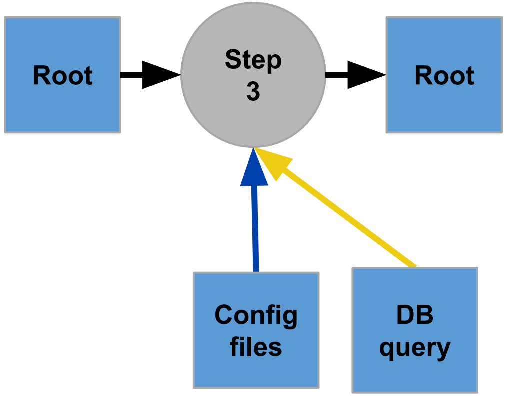
CMS workflow I/O: Step 2 (DIGI/RAW): Overlay pile-up hits and digitization of hits



Simulation of digitization of detector response to simulated hits (and overlayed simulated hits from pile-up events) to produce simulated detector output and simulation of Level 1 trigger

- Inputs: GEN/SIM ROOT, P.U. ROOT, config files, DB query
- Outputs: RAW data in ROOT file

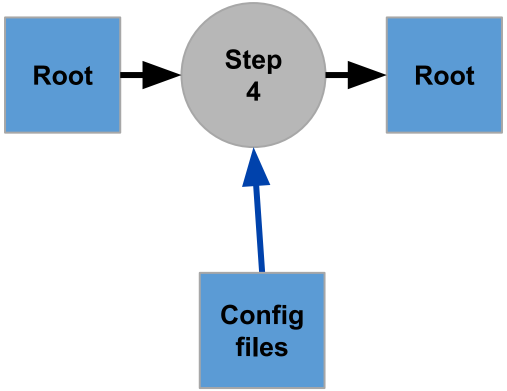
CMS workflow I/O: Step 3: Event reconstruction



Reconstruction of real or simulated detector raw data into physics objects

- Inputs: DIGI/RAW ROOT, config files, DB query
- Outputs: Physics objects in ROOT file

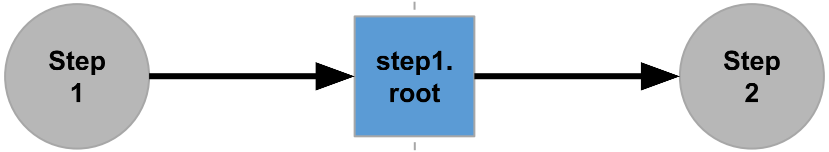
CMS workflow I/O: Step 4: Event selection and Ntuple creation



Event selection and Ntuple creation

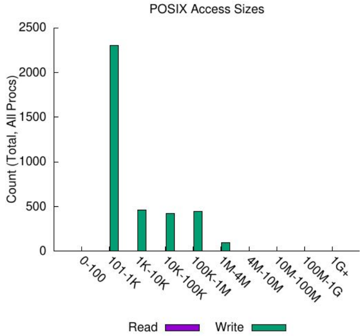
- Inputs: Physics object ROOT, config files
- Outputs: Ntuples in ROOT file

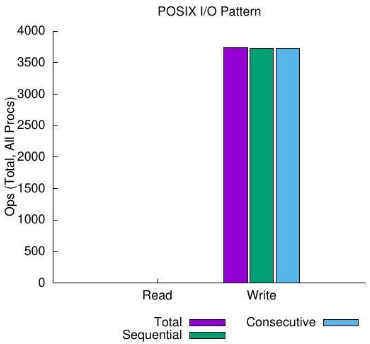
CMS workflow I/O: Steps 1 & 2

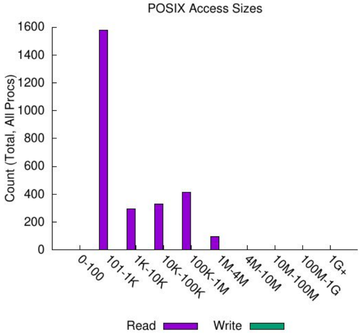


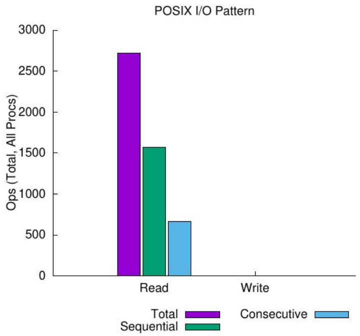
~370 MB written
~880 MB/sec
~8,900 IOPs/sec

~360 MB read
~2,100 MB/sec
~16,000 IOPs/sec

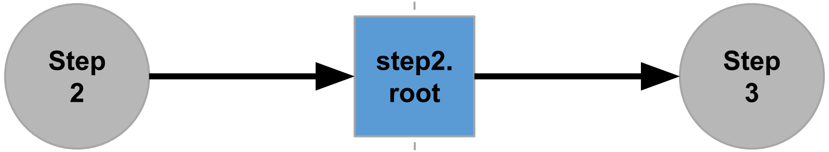




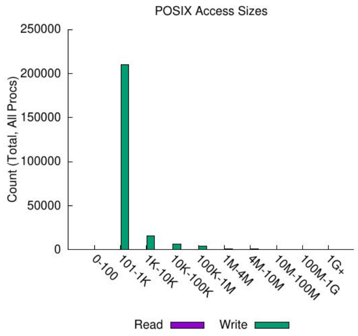


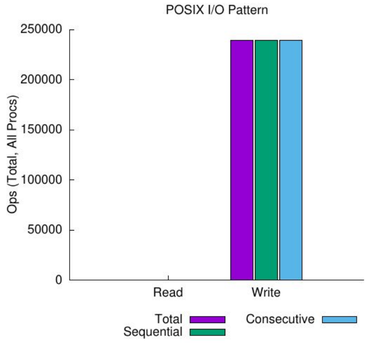


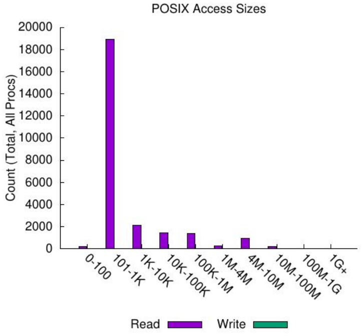
CMS workflow I/O: Steps 2 & 3

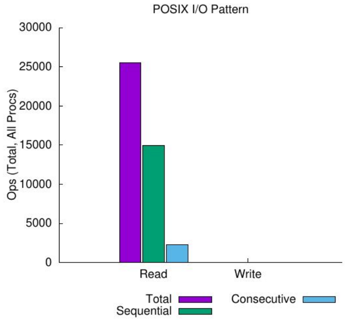


~13,500 MB written	**~10,200 MB read**
~1,100 MB/sec	**~2,400 MB/sec**
~20,000 IOPs/sec	**~6,000 IOPs/sec**

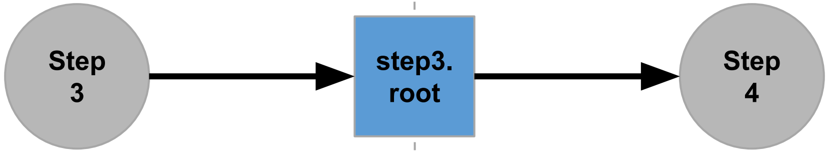






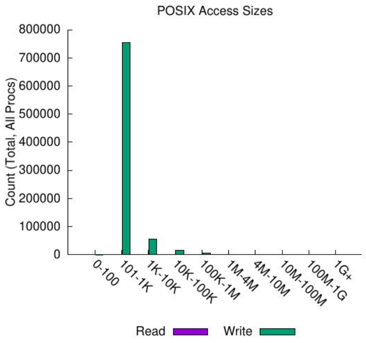


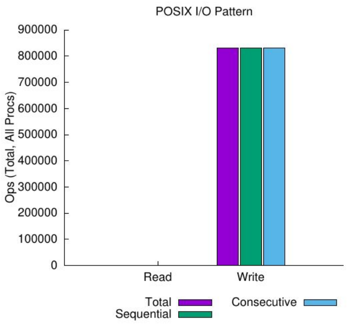
CMS workflow I/O: Steps 3 & 4

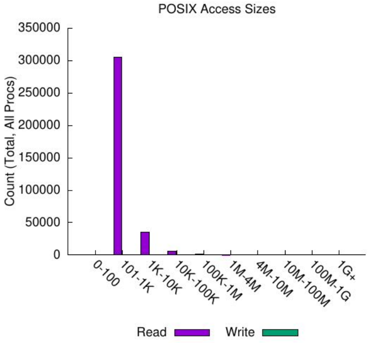


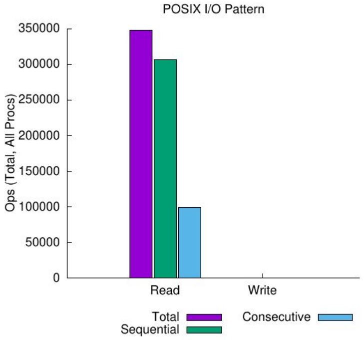
~4,500 MB written
~310 MB/sec
~58,000 IOPs/sec

~844 MB read
~715 MB/sec
~300,000 IOPs/sec

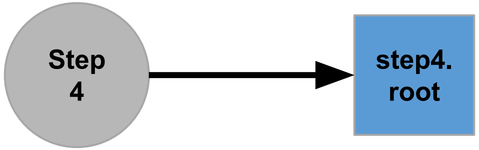








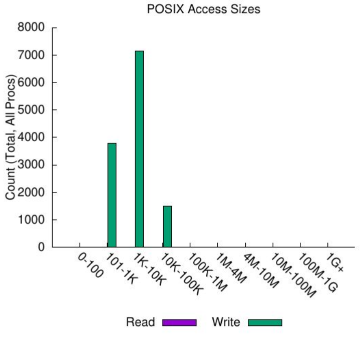
CMS workflow I/O: Step 4 final output

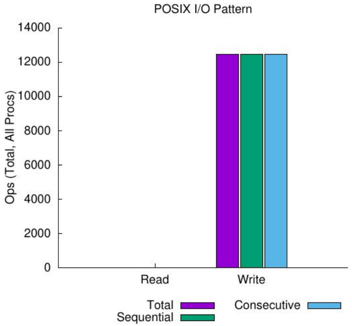


~50 MB written

~150 MB/sec

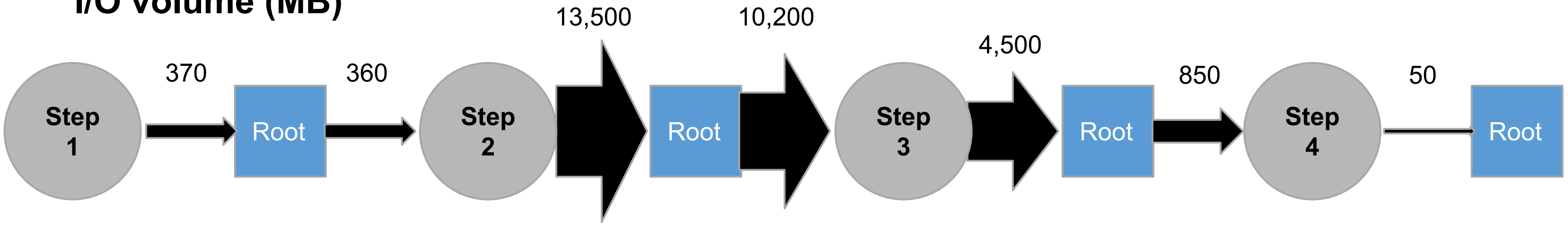
~40,000 IOPs/sec



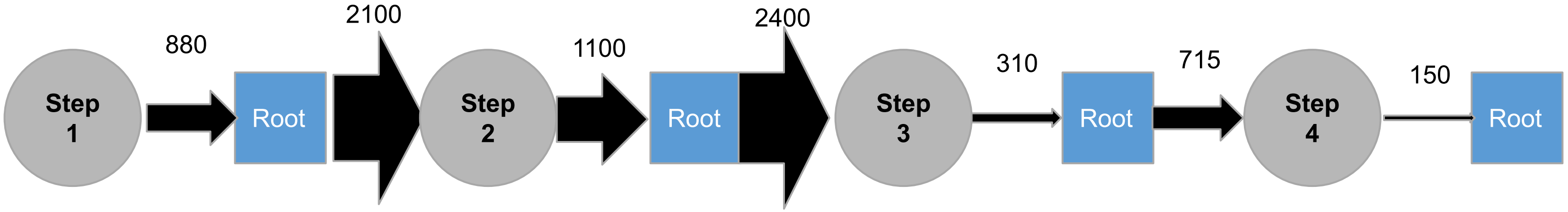


CMS workflow I/O: Step 4 final output

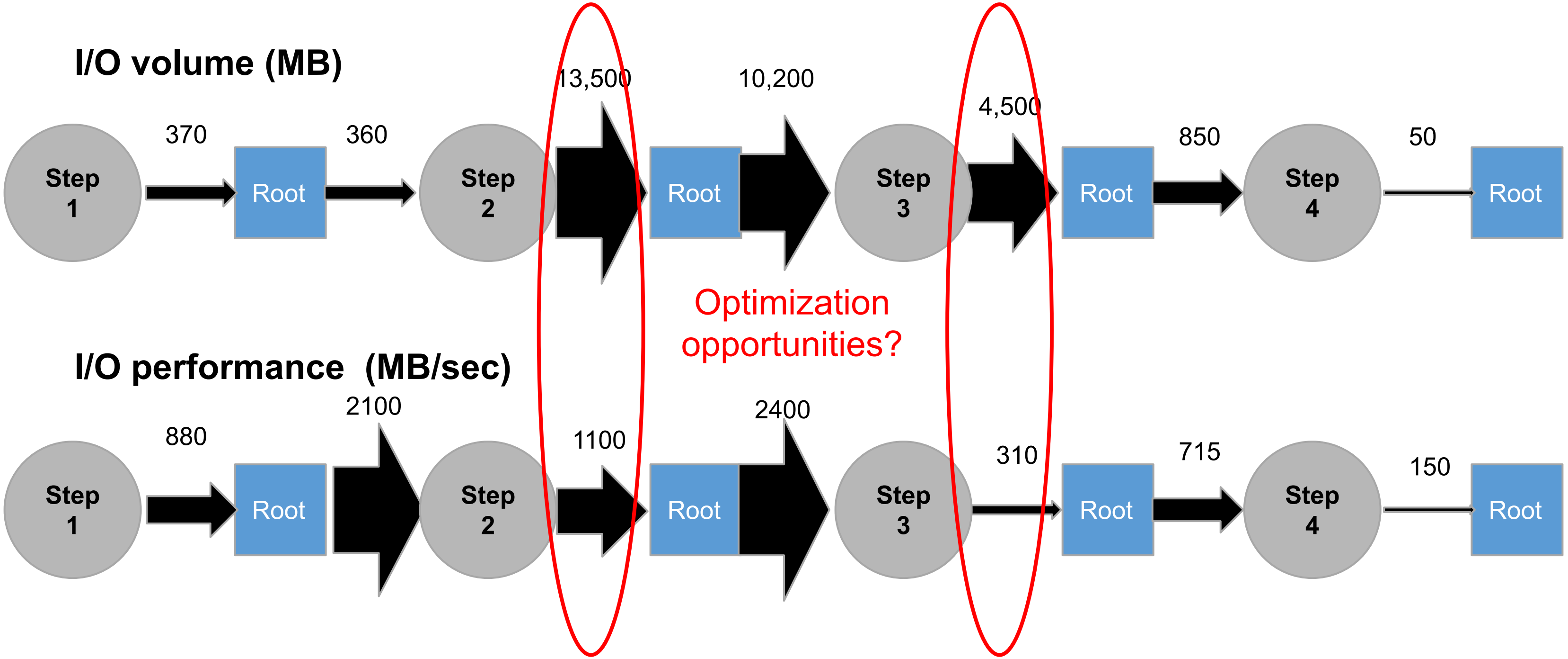
I/O volume (MB)



I/O performance (MB/sec)



CMS workflow I/O: Step 4 final output



Darshan analysis of DUNE

DUNE workflow I/O analysis

- Similar in nature to CMS workflow pattern described previously (i.e., event generation + detector simulation + reconstruction)
  - Similar dependence on technologies like cvmfs, XRootD, etc.

- Preliminary Darshan results for standard reconstruction of ProtoDUNE-SP raw data on DUNE interactive machines
  - Raw inputs read from NFS for these tests (rather than traditional XRootD reading of data)
  - Run 5 events (input file contains O(100) events)

File Count Summary
(estimated by POSIX I/O access offsets)

type	number of files	avg. size	max size
total opened	467	17M	7.7G
read-only files	171	98K	11M
write-only files	3	72K	89K
read/write files	10	791M	7.7G
created files	13	609M	7.7G

Data Transfer Per Filesystem (POSIX and STDIO)

File System	Write		Read	
	MiB	Ratio	MiB	Ratio
/cvmfs/dune.opensciencegrid.org	0.29843	0.46604	81.71731	0.04658
/cvmfs/larsoft.opensciencegrid.org	0.13174	0.20573	1672.43887	0.95342
UNKNOWN	0.14645	0.22871	0.00000	0.00000
/grid/fermiapp	0.00000	0.00000	0.00000	0.00000
/	0.00000	0.00000	0.00000	0.00000
/dune/data	0.06373	0.09952	0.00000	0.00000

DUNE workflow I/O analysis

- Similar in nature to CMS workflow pattern described previously (i.e., event generation + detector simulation + reconstruction)
  - Similar dependence on technologies like cvmfs, XRootD, etc.

- Preliminary Darshan results for standard reconstruction of ProtoDUNE-SP raw data on DUNE interactive machines
  - Raw inputs read from NFS for these tests (rather than traditional XRootD reading of data)
  - Run 5 events (input file contains O(100) events)

File Count Summary
(estimated by POSIX I/O access offsets)

type	number of files	avg. size	max size
total opened	467	17M	7.7G
read-only files	171	98K	11M
write-only files	3	72K	89K
read/write files	10	791M	7.7G
created files	13	609M	7.7G

Note input file is read/write (and also mistakenly labeled as created by Darshan)

Accessed up to offset of 7.7 GB into file

Data Transfer Per Filesystem (POSIX and STDIO)

File System	Write		Read	
	MiB	Ratio	MiB	Ratio
/cvmfs/dune.opensciencegrid.org	0.29843	0.46604	81.71731	0.04658
/cvmfs/larsoft.opensciencegrid.org	0.13174	0.20573	1672.43887	0.95342
UNKNOWN	0.14645	0.22871	0.00000	0.00000
/grid/fermiapp	0.00000	0.00000	0.00000	0.00000
/	0.00000	0.00000	0.00000	0.00000
/dune/data	0.06373	0.09952	0.00000	0.00000

Total read volume of around 1.6 GB, confirming partial access of input dataset

DUNE workflow I/O analysis

- Similar in nature to CMS workflow pattern described previously (i.e., event generation + detector simulation + reconstruction)
  - Similar dependence on technologies like cvmfs, XRootD, etc.

- Preliminary Darshan results for standard reconstruction of ProtoDUNE-SP raw data on DUNE interactive machines
  - Raw inputs read from NFS for these tests (rather than traditional XRootD reading of data)
  - Run 5 events (input file contains O(100) events)

File Count Summary
(estimated by POSIX I/O access offsets)

type	number of files	avg. size	max size
total opened	467	17M	7.7G
read-only files	171	98K	11M
write-only files	3	72K	89K
read/write files	10	791M	7.7G
created files	13	609M	7.7G

Data Transfer Per Filesystem (POSIX and STDIO)

File System	Write		Read	
	MiB	Ratio	MiB	Ratio
/cvmfs/dune.opensciencegrid.org	0.29843	0.46604	81.71731	0.04658
/cvmfs/larsoft.opensciencegrid.org	0.13174	0.20573	1672.43887	0.95342
UNKNOWN	0.14645	0.22871	0.00000	0.00000
/grid/fermiapp	0.00000	0.00000	0.00000	0.00000
/	0.00000	0.00000	0.00000	0.00000
/dune/data	0.06373	0.09952	0.00000	0.00000

Some open questions:

1. Write activity to cvmfs, which is read-only?
2. DUNE output files not currently captured by Darshan?

Next steps

- Investigate and develop workarounds for numerous obstacles to instrumenting ROOT I/O workloads
  - Unreliable I/O instrumentation for forked processes
  - Difficulty deploying Darshan in complex production environments (i.e., workflows employing containers)
  - Missing Darshan log data (e.g., due to instrumenting too many workflow files)
- With broader instrumentation coverage of ATLAS, CMS, DUNE workflows, we can turn focus to Darshan analysis tools to better understand flow of data, I/O access patterns, and achieved performance of different HEP data processing stages
  - Drive tuning decisions for ROOT usage and for ROOT implementation
  - New Python bindings for Darshan log utility library should ease development of new analysis tools

Thanks!